Nonlinear Network Time-Series Forecasting Using Redundancy Reduction

Pramod Lakshmi Narasimha, Michael T. Manry and **Changhua Yu**
Department of Electrical Engineering, University of Texas at Arlington.
l_pramod@uta.edu, manry@uta.edu

Abstract

In this paper we propose an efficient method for forecasting highly redundant time-series based on historical information. First, redundant inputs and desired outputs are compressed and used to train a single network. Second, network output vectors are uncompressed. Our approach is successfully tested on the hourly temperature forecasting problem.

Introduction

Time-series forecasting is a very important problem in many disciplines. Correct trend prediction, can be used profitably in stock market analysis (Saad, Prokhorov, & (II) 1998). Power load forecasting is critically important for electric utility companies (Hippert, Pedreira, & Souza 2001). River flow forecasting can have significant economic impact in agricultural water management and in protection from water shortages and possible flood damage (Atiya *et al.* 1999).

Most neural net time-series forecasters use the highly redundant time domain sequence as the network inputs (Khotanzad *et al.* 1997), (Ling *et al.* 2003) resulting either in large dimensionality of the input space or in a large number of networks.

In this paper, we describe a neural net time series forecaster that uses data compression to reduce redundancy. We present a method for compressing the temperature data and derive bounds on the training error in the compressed domain.

Data Organization

For electric utilities, temperature forecasts are critical. Modeling the temperature variations of a region with a small temperature data set is an important issue. If each pattern starts at the same time (*i.e.*, time corresponding to the first temperature input), then, the maximum number of patterns ($N_{v_{24}}$) would be 365 per year. We can increase the number of patterns by having them start at different times, that is having a lower *step size* (SS).

The temperature inputs are represented by vectors $\{\mathbf{x}_p \in \mathbb{R}^N, p = 1, 2, \cdots, N_v\}$, which contain previous 24 hourly temperatures. The forecast high and low temperatures of the next 24 hours are represented by a vector $\mathbf{\Psi}_p$. The time related inputs can be represented by a vector $\mathbf{\Theta}_p$. The desired outputs are represented by the vector $\{\mathbf{t}_p \in \mathbb{R}^M, p = 1, 2, \cdots, N_v\}$, which contains the next day's hourly temperatures. Here N_v is the number of patterns. The format of the training file is $\{\mathbf{x}_{\mathbf{a}p}, \mathbf{t}_p\}_{p=1}^{N_v}$, where the augmented input vector is

$$\mathbf{x}_{\mathbf{a}p} = [\mathbf{x}_p^T, \mathbf{\Psi}_p^T, \mathbf{\Theta}_p^T]^T \quad (1)$$

Redundancy Reduction

We can eliminate temperature sequence redundancy by applying *Karhunen Loéve Transform* (KLT).

The KLTs of $\mathbf{x}_p$ and $\mathbf{t}_p$ are denoted by $\mathbf{X_p}$ and $\mathbf{T_p}$. We use the first N_k coefficients of the transformed vectors $\mathbf{X}_p$'s and M_k coefficients of the transformed vectors $\mathbf{T}_p$'s for training.

Using the triangular inequality, we derive a bound on the training error in the compressed domain,

$$\|\mathbf{T}_p - \mathbf{Y}_p\| = \|\mathbf{t}_{\mathbf{r}p} - \mathbf{y}_{\mathbf{r}p}\| \leq \|\mathbf{t}_{\mathbf{r}p} - \mathbf{t}_p\| + \|\mathbf{t}_p - \mathbf{y}_{\mathbf{r}p}\| \quad (2)$$

where $\mathbf{T}_p$ and $\mathbf{Y}_p$ are the desired and forecast temperatures in compressed domain, $\mathbf{t}_{\mathbf{r}p}$ and $\mathbf{y}_{\mathbf{r}p}$ are the reconstructed time domain temperatures from $\mathbf{T}_p$ and $\mathbf{Y}_p$ respectively, $\mathbf{t}_p$ is the desired temperature vector in the data domain. It is observed that the reconstruction error ($\|\mathbf{t}_{\mathbf{r}p} - \mathbf{t}_p\|$) is usually very small. Equality in 2 occurs when there is no compression.

Multilayer Perceptron

Training and Sizing

A schematic of the forecasting system is shown in the Figure 1. We use the Output Weight Optimization - Hidden Weight Optimization (OWO - HWO) algorithm described in Yu & Manry (2002) for training.

We use the *structural risk minimization* principle to choose the number of hidden units (N_h). It is clearly observed from Figure 2 that the test error for three hidden units is the minimum for data set North of Table 1. Hence the number of hidden units for that set of training data is chosen to be three and the corresponding network is saved.

System Testing

In Table 1, the values of mean absolute error (MAE) and standard deviation of error (SDE) of our proposed system is compared with the time domain system.

*We would like to convey our thanks to Dr. Alireza Khotanzad for providing the data sets used here.

†This work was supported by the Advanced Technology Program of the state of Texas, under grant number 003656-0129-2001.

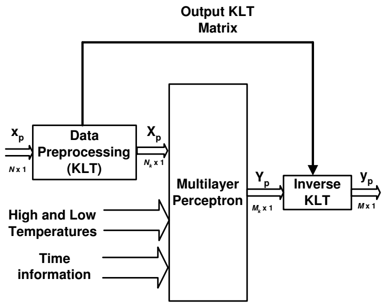


Figure 1: Forecasting System Diagram

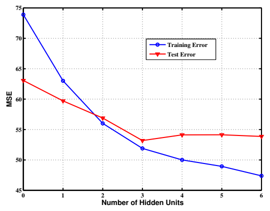


Figure 2: Training and test errors versus N_h

Figure 3 shows plots of MAE and SDE for training and testing (SS = 4). In Figure 4, we show examples of actual temperatures and forecasts. These forecasts used high and low temperatures of the day following the forecast day as inputs for training. Network sizes for our system and time-domain forecasting system are summarized in Table 2. From the plots, the proposed system works quite well.

Table 1: MAE and SDE of two systems for SS = 4.

	Our system		Time domain system	
Region	MAE	SDE	MAE	SDE
Coast	0.78	1.11	1.53	2.16
East	1.17	1.54	2.53	3.27
Far West	1.04	1.48	7.75	9.50
North	0.85	1.19	5.91	7.42
North Central	0.74	1.07	5.68	7.05
South	0.66	0.94	4.83	6.19
South Central	0.84	1.25	5.87	7.49
West	1.59	1.20	2.51	8.25
Average Error	0.86	1.22	5.09	6.42

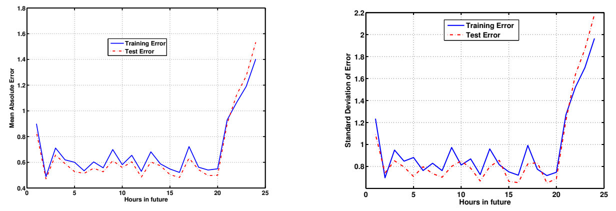
References

Atiya, A. F.; El-Shoura, S. M.; Shaheen, S. I.; and El-Sherif, M. S. 1999. A comparison between neural-network forecasting techniques-case study: River flow forecasting. *IEEE Trans. Neural Networks* 10(2):402–409.

Chen, H.-H., and Manry, M. T. 1996. A neural network training algorithm utilizing multiple sets of linear equations. In *The 30th Asilomar Conference on Signals, Systems and Computers*, volume 2, 1166–1170.

Hippert, H. S.; Pedreira, C. E.; and Souza, R. C. 2001. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Trans. Power Systems* 16(1):44–55.

Khotanzad, A.; Davis, M. H.; Abaye, A.; and Maratukulam, D. J. 1996. An artificial neural network hourly temperature forecaster with applications in load forecasting. *IEEE Trans. on Power Systems* 11(2):870–876.



(a) (b)

Figure 3: (a) MAE, and (b) SDE (SS = 4).

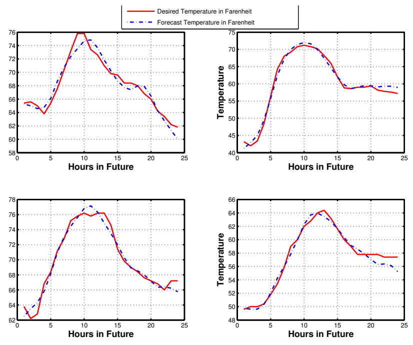


Figure 4: Plot of desired and predicted temperatures for our forecaster for some sample days (SS = 4). (a) Jul 12,1998, (b) Aug 24, 1998, (c) Sep 27, 1998, (d) Oct 19, 1998

Khotanzad, A.; Afkhami-Rohani, R.; Tsun-Liang, L.; Abaye, A.; Davis, M.; and Maratukulam, D. J. 1997. Annstlf-a neural-network-based electric load forecasting system. *IEEE Trans. on Neural Networks* 8(4):835–846.

Ling, S. H.; Leung, F. H. F.; Lam, H. K.; and Tam, P. K. S. 2003. Short-term electric load forecasting based on a neural fuzzy network. *IEEE Trans. on Industrial Electronics* 50(6):1305–1316.

Saad, E. W.; Prokhorov, D. V.; and (II), D. C. W. 1998. Comparitive study of stock trend prediction using time delay, recurrent and probalilistic neural networks. *IEEE Trans. Neural Networks* 9(6):1456–1468.

Yu, C., and Manry, M. T. 2002. A modified hidden weight optimization algorithm for feed-forward neural networks. In *The 36th Asilomar conference on Signals, Systems and Computers*, volume 2, 1034–1038.

Table 2: Network size for data set South.

	Our system		Time domain system	
	SS = 24	SS < 24	SS = 24	SS < 24
N	12	14	28	30
M	6	6	24	24
N_h	5	5	5	5